

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Дисциплина «Методы программирования»

Лабораторная работа №6

на тему «Деревья: двоичное дерево поиска, обходы дерева,

дерево отрезков, декартово дерево»

Выполнила студент группы ВКБ31

Закарян Т.Г.

г. Ростов-на-Дону

2021 г.

Задача Декартово дерево

Код программы:

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

struct edge{

int from, to, cost;

};

const int INF = 1e9;;

int main(){

int n;

cin » n;

vector <edge> E;

for(int i = 0; i < n; i++){

for(int j = 0; j < n; j++){

int x;

cin » x;

if(x != 0 && x != 100000){

E.push\_back({i,j,x});

}

}

}

int x ;

vector<int> d(n, INF), p(n, -1);

d[0] = 0;

for(int i = 0; i < n; i++){

x = -1;

for(int j = 0; j < E.size(); j++){

int from = E[j].from;

int to = E[j].to;

int cost = E[j].cost;

if(d[to] > d[from] + cost ){

d[to] = max(d[from] + cost, -INF);

p[to] = from;

x = to;

}

}

}

if(x == -1){

cout « "NO" « endl;

}else{

int y = x;

for(int i = 0; i < n; i++){

y = p[y];

}

vector <int> path;

for(int cur = y;; cur = p[cur]){

path.push\_back(cur);

if(cur == y && path.size() > 1){

break;

}

}

reverse(path.begin(), path.end());

cout « "YES" « endl;

cout « path.size() « endl;

for(int i = 0; i < path.size(); i++){

cout « path[i] + 1;

if(i != path.size()-1){

cout « ' ';

}

}

cout « endl;

}

return 0;

}

Оценка программы:

